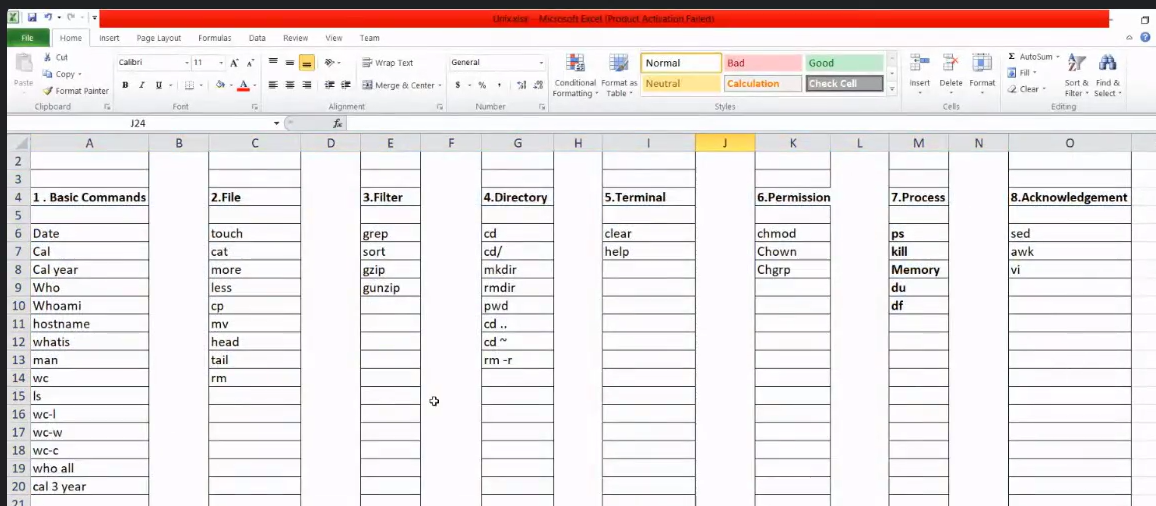


Unix is a modular OS made up of a number of essential components, including the kernel, shell, file system and a core set of utilities or programs. At the heart of the Unix OS is the kernel, a master control program that provides services to start and end programs.

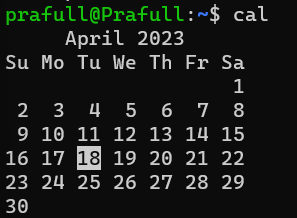


**\*\*BASIC COMMANDS\*\***

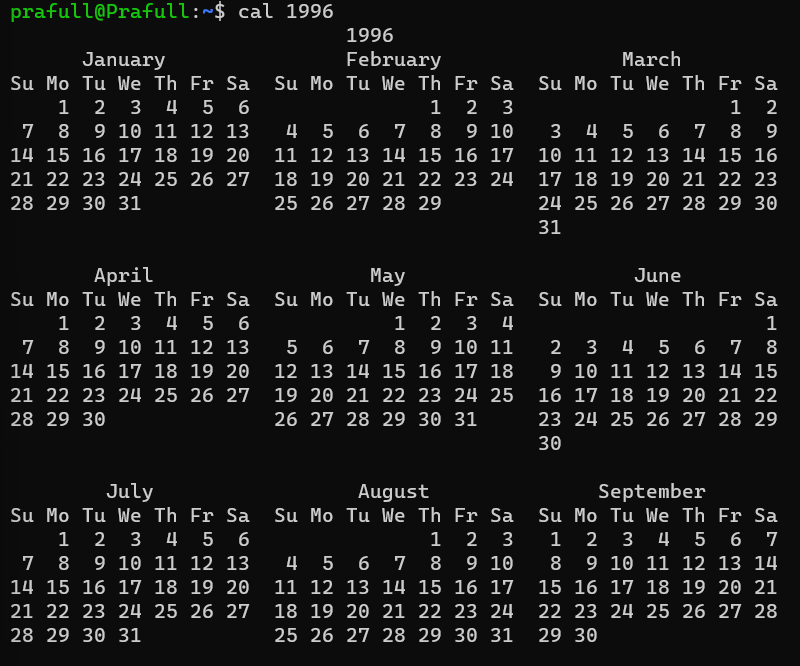
1.date-it will show today’s date and time



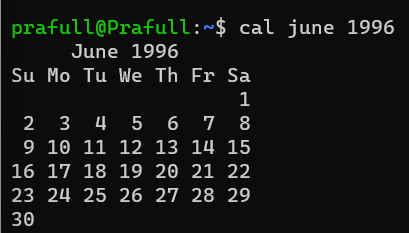
2.a) cal -show’s current month + today’s date highlighted.



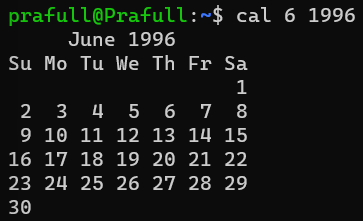
b)cal year – show’s specific year’s whole calendar



c) cal month year - show’s specific year’s whole calendar with given month



d) cal 6 year -same as above but here we use number instead of month name



3.who – It will show the only currently login users available on same network.

4.who -all :–(who + space +hyphen +all)[Always use hyphen work in interview]

All login + logout users

5.whoami-personal/device user name:-



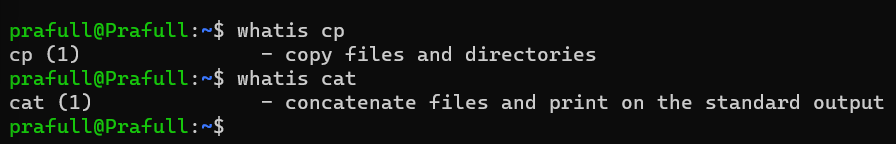
6.a) hostname: - device name/username/computer device name



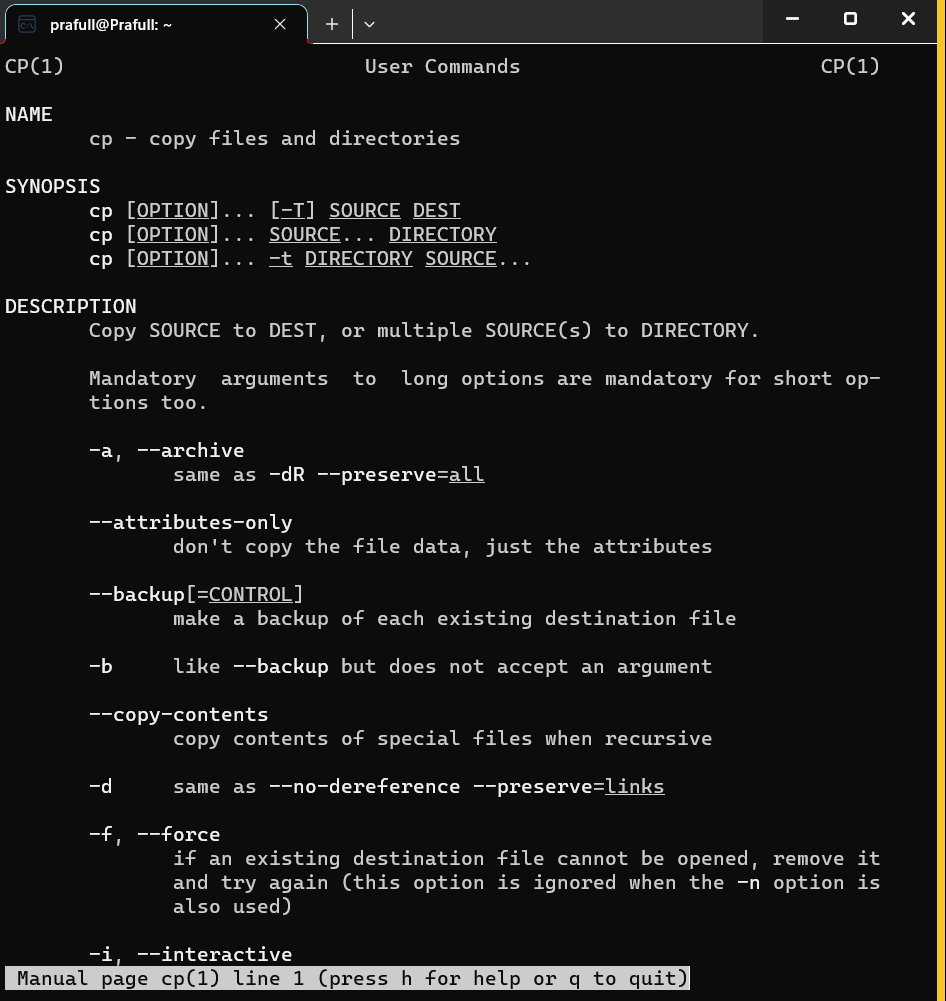
b) hostname -i : - ip address of connected network mostly same for everyone who are connected on that network.



7.whatis :- it gives the one line description of a command.



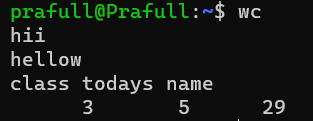
8.man :- it gives brief description of commands (press ctrl Q to exit)



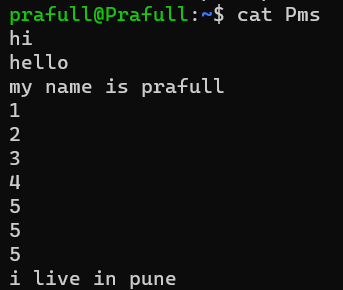
9.ls :- shows no. of files & folders created on device network

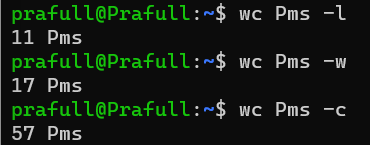


10.wc :- shows no. of lines, words & characters(include space + enter key)



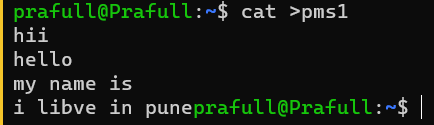
1. wc filename -l (no. of lines)
2. ) wc filename -w (no. of words)
3. ) wc filename -c (no. of character)

data of Pms file

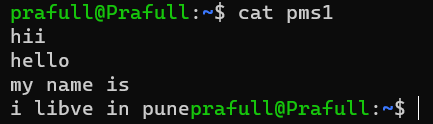
Pms-filename

**\*\* File related commands \*\***

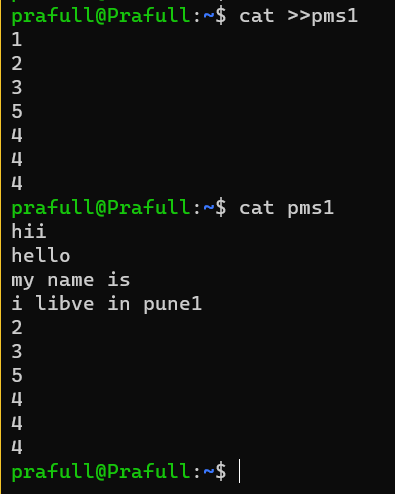
1. cat(to create ,display & edit file)
2. cat >pms1 :- [cat >filename]{create + write}



b.cat pms1 :- [cat filename]{display file}



1. cat >>pms1 :- [cat >>filename]{To edit file but the data we insert in this command always add at the end of previous data}

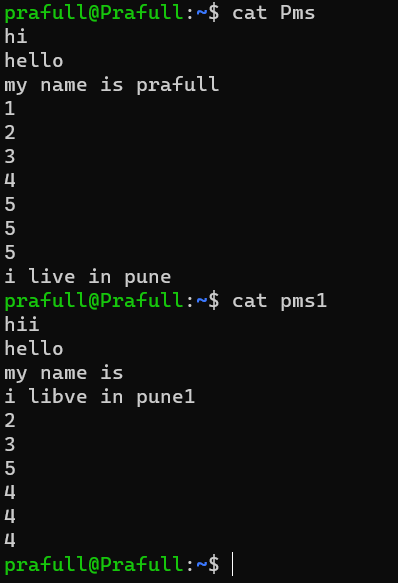


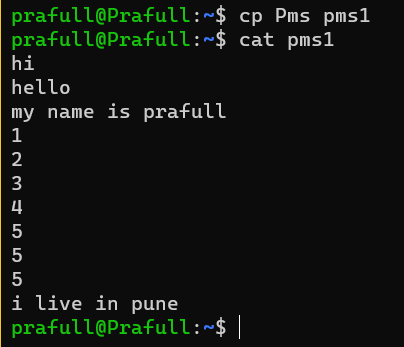
2.touch filename :- simply just to create a file no data will be added



3.cp filename1 filename2: -(filename1 remains empty and available in storage)

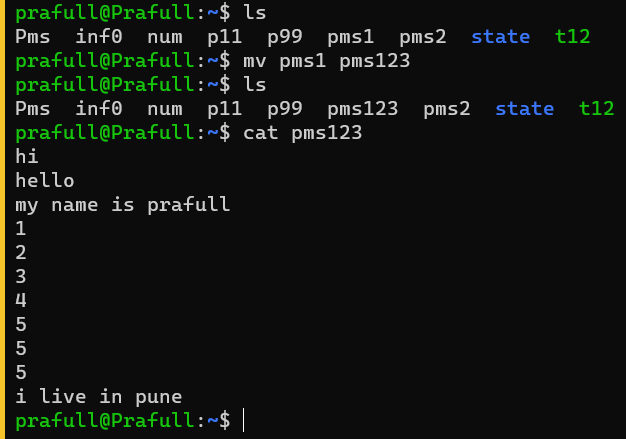
[copy + paste , from filename1 to filename2, but if there is any data available in filename2 it will override that data ,if filename2 this file does not exist in storage it will create filename2 and paste the data in it.]





4. mv filename1 filename2: -(filename1 is deleted and not available in storage)

[cut + paste , from filename1 to filename2, but if there is any data available in filename2 it will override that data ,if filename2 this file does not exist in storage it will create filename2 and paste the data in it.]

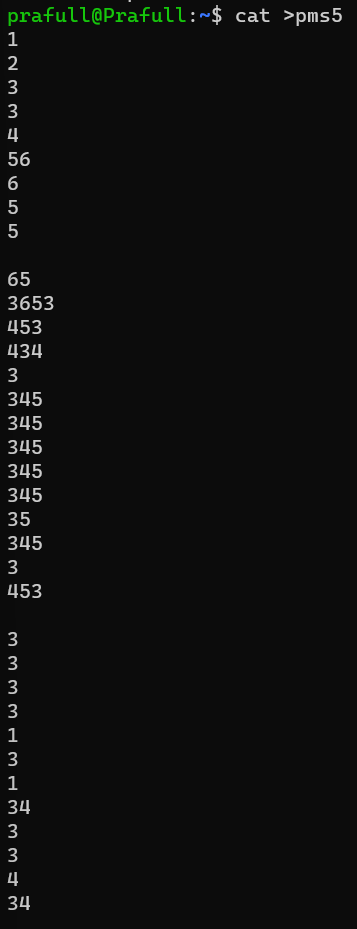


Pms1 file is deleted permanently.

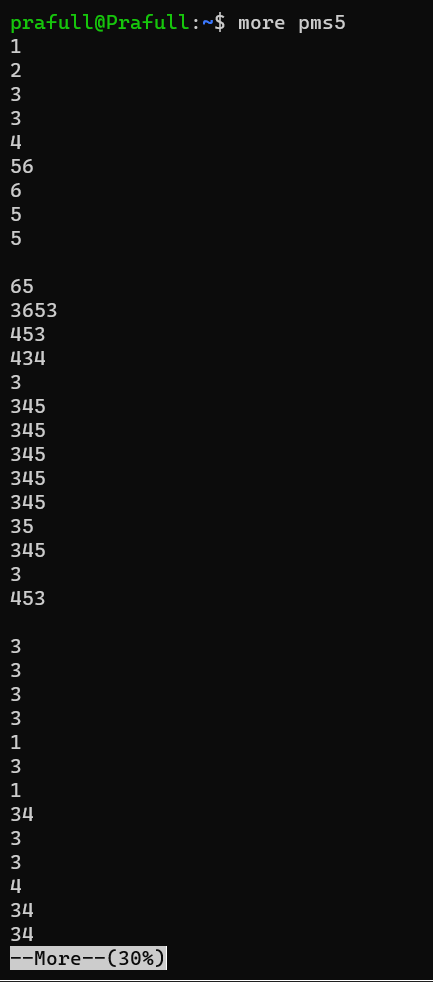
5.more filename: -(spacebar is used to next page)

{Page by page operation}

[if the file is having no. of line (ex.1000)so every time we open the file we need to scroll down up to1000 line every time , more command breakdown this type of file into pages ]

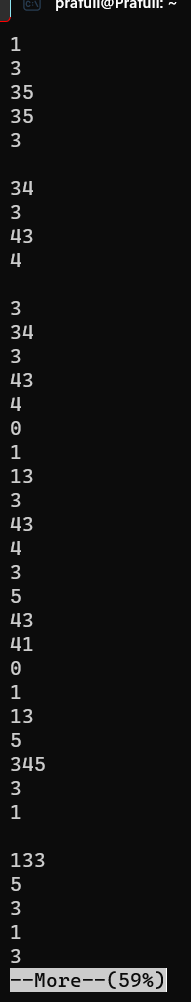
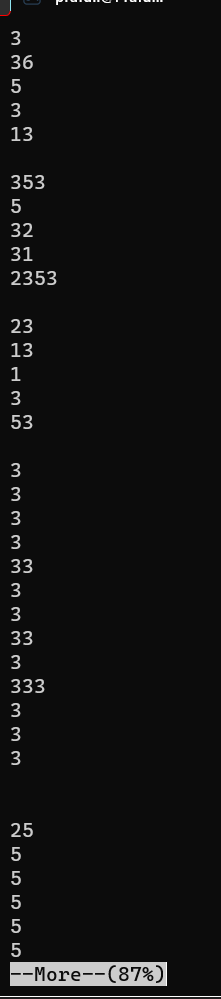
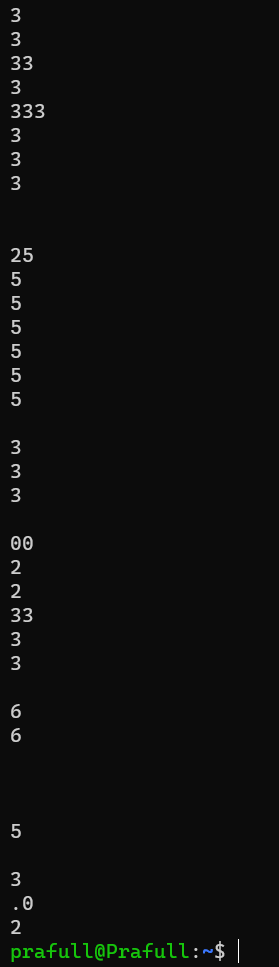


After command Page 1



Press spacebar for next page.

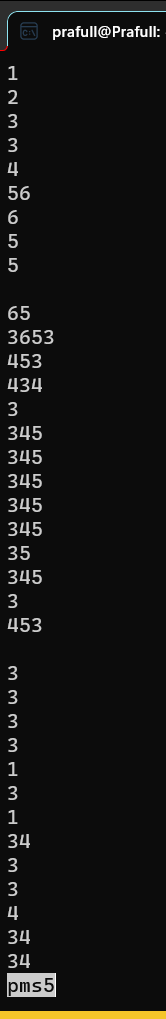
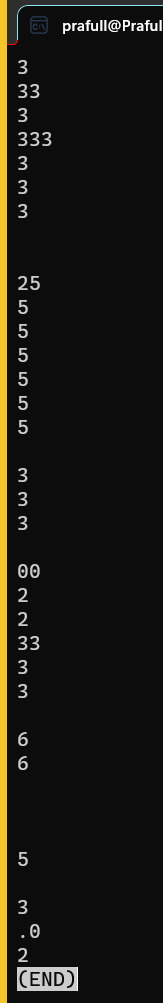
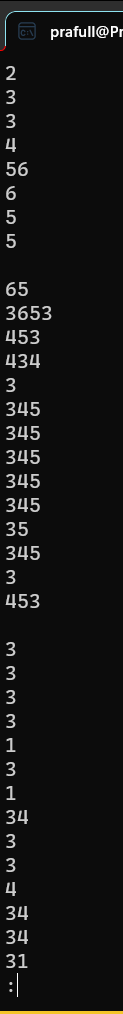
Page 2 Page 3 Page 4

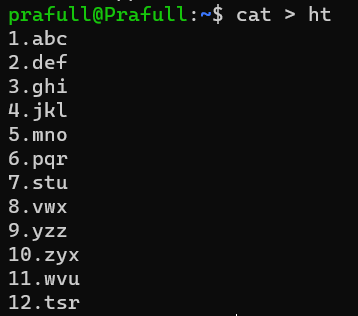
6.less filename: -(spacebar is used to next page)(ctrl X to stop)

[below to above){line by line}

[if the file is having no. of line (ex.1000)so every time we open the file we need to scroll down up to1000 line every time , more command breakdown this type of file into pages ]

7.head and tail



Here 1 file is created

1. If we want to display from line 3 to 10 there is one formula is applied

Upper limit – lower limit + 1

=10(Upper limit of line no.) – 3(lower limit of line ) + 1

=8

For command

head is upper limit : - head -10

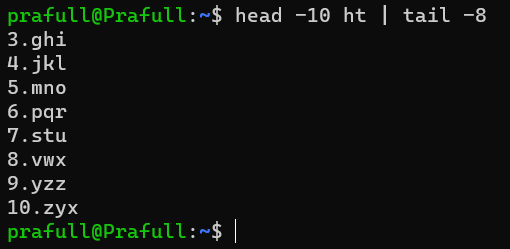
tail is the answer of above :- tail -8

Working: - as head is 10 so it goes to line 10(head) for start then count 8(tail) in upward direction including no. 10 row and it stops at line 3.

So it displays line 3 to 10.

Command :-

head -10 filename | tail 8

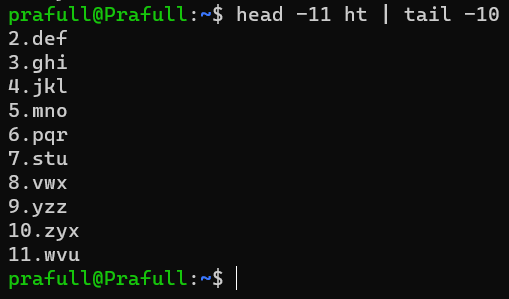


1. For line 2 to 11: -

11(upper limit) – 2(lower limit) + 1

10

head-11 ; tail -10

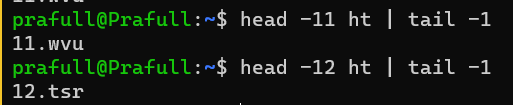


1. Display line no. 12 only

12(upper limit) – 12(lower limit) + 1

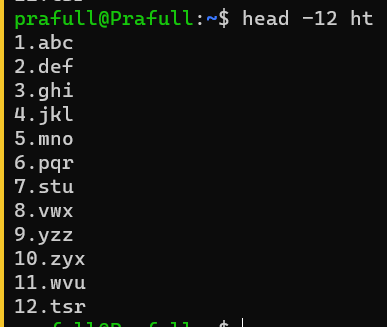
=1

head -12 | tail -1

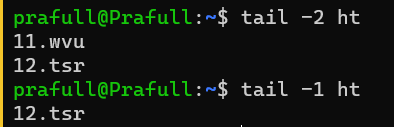


Line no. 11 also displayed.

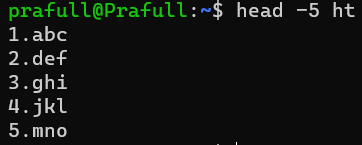
1. Line 1 to 12



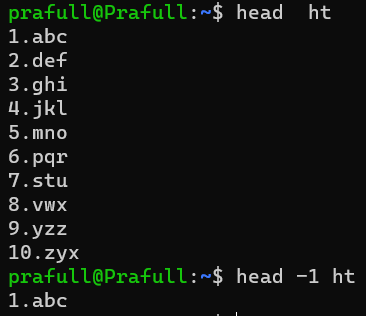
1. Bottom 2 lines / 1 line



1. 1st 5 lines



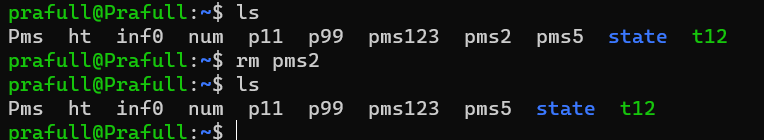
1. 1st line



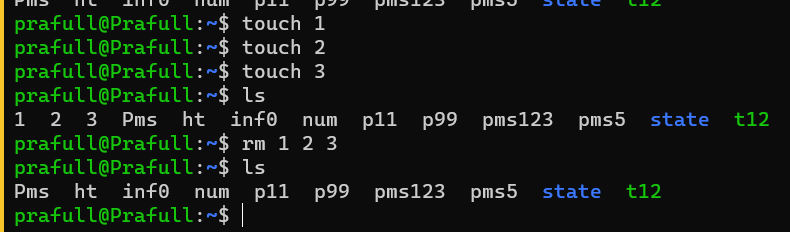
8:- rm command to delete single also multiple files

rm filename1 filename2

Single file delete



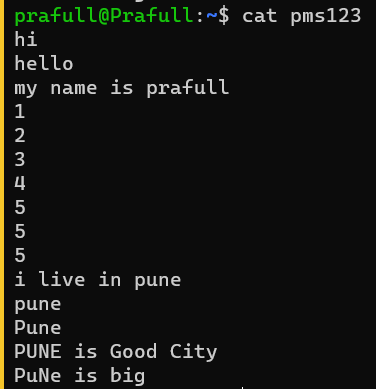
Multiple files delete.

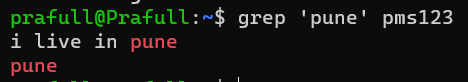


**\*\*Filter related commands\*\***

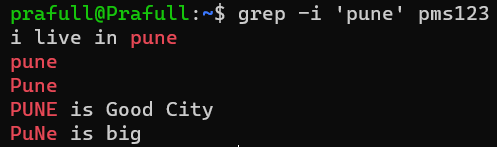
1. A) grep ‘word’ filename :- return all the lines of data in which ‘word’ is available

It is case sensitive .

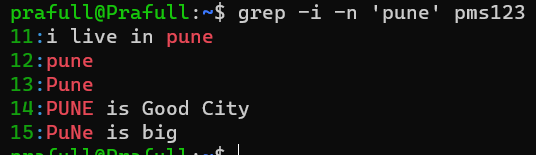




B). grep -i ‘word’ filename – for non-case sensitive result.



C) grep -i -n ‘word’ filename – To get line no. where that word is present.

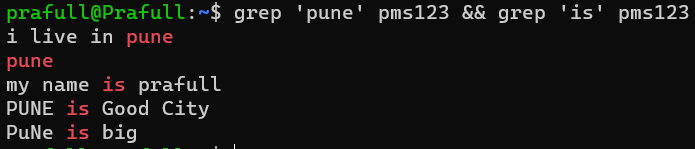


1. grep -i -c ‘word' filename - To get total no. of lines where that word is present.

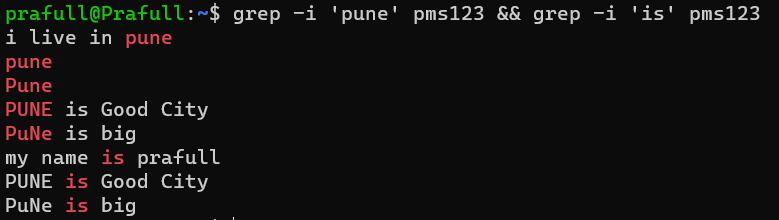
****

**FOR MULTIPLE FILTER**

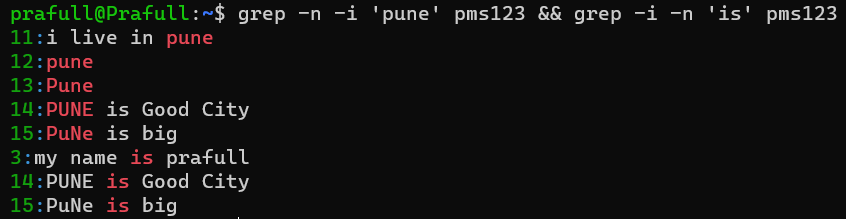
a.1) grep ‘word1’ filename && grep ‘word2’ filename



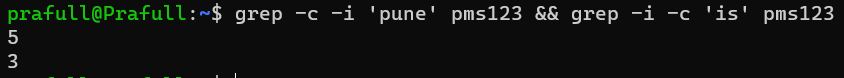
b.1) grep -i ‘word1’ filename && grep -i ‘word2’ filename



c.1) grep -i -n ‘word1’ filename && grep -i -n ‘word2’ filename

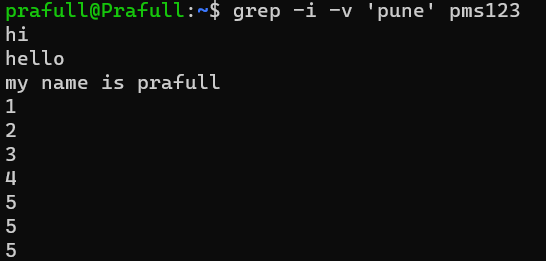


d.1) grep -i -c ‘word1’ filename && grep -i -c ‘word2’ filename



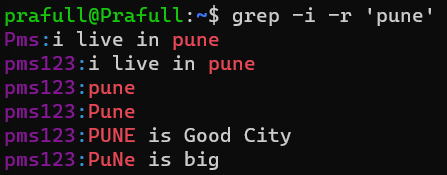
**\*\* EXCLUDE \*\***

1. grep -i -v ‘word’ filename

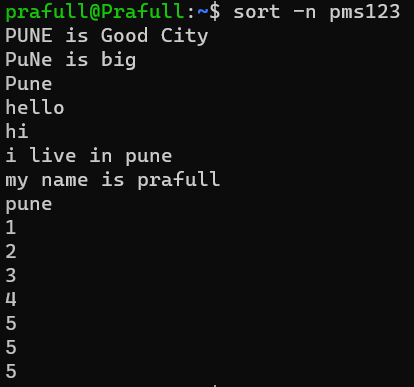


**History of that word**

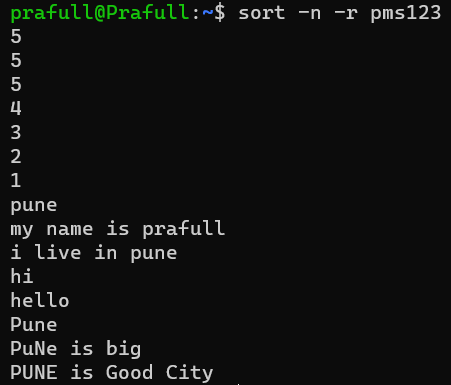
1. grep -i -r ‘word’ filename



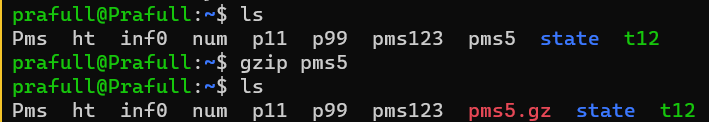
1. a.) sort -n filename – Ascending order :- CAPITAL LETTER -small letters-(1,2,3,4)numbers



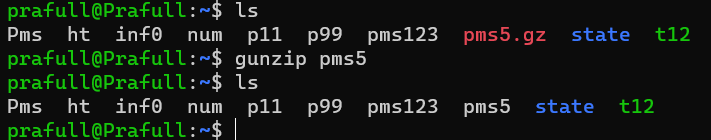
b.) sort -n -r filename – Descending order-numbers-small letter-CAPITAL LETTER



1. gzip filename – compress file & save in zip format

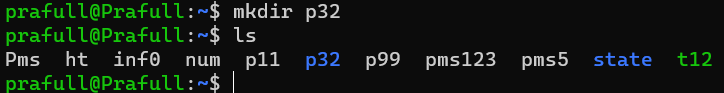


1. gunzip filename - Decompress file & save in original format

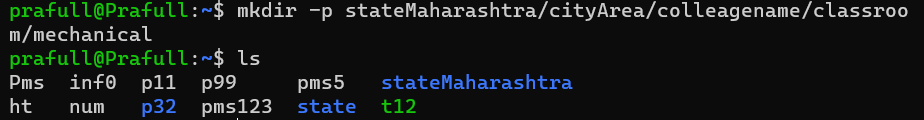


**\*\*Directory (folder) related commands\*\***

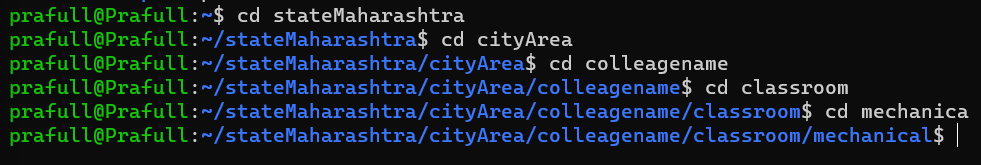
1. mkdir -make directory -to create directory/folder



1. mkdir -p stateMaharashtra/cityArea/colleagename/classroom/mechanical – To create main dir with sub dir



1. a. cd dir-name- change directory- Here we change dir one by one

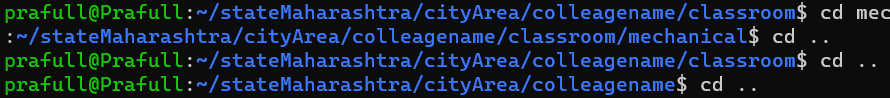


b.direct jump into any dir:

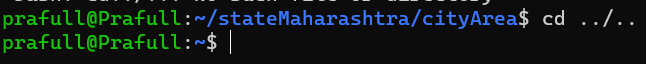


1. cd . . – one by one back from dir





1. cd . . / . .

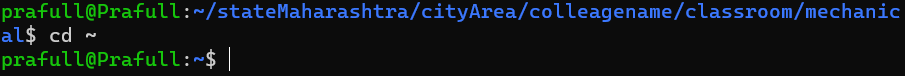


Multiple dir back



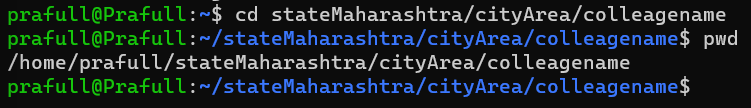
Directly go outside of all dir

Cd ~

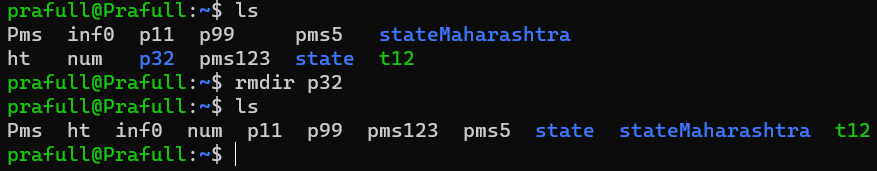


1. pwd :- present working dir

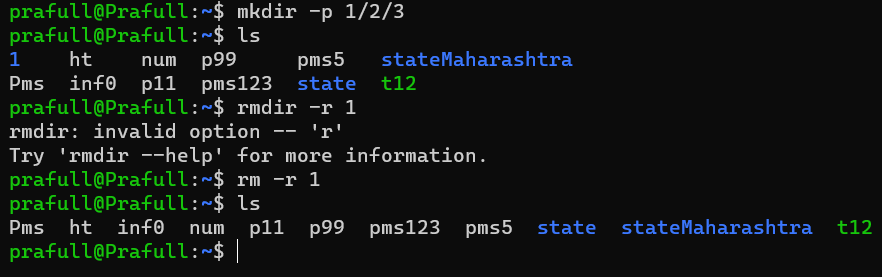




1. rmdir filename – but here dir needs to be empty then and then it will delete.



1. rm -r dirname – it will delete the non-empty dir.



**\*\* Permission Related Commands \*\***

r-read permission-4

w-write permission-2

x – execute permission-1

Rules for-

Ex. If there is 3 groups in that Team

5 developers – A B C D E

3 tester -

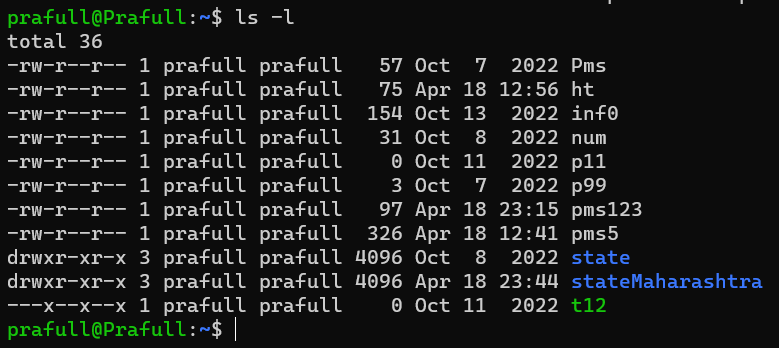
4ba - X Y Z S

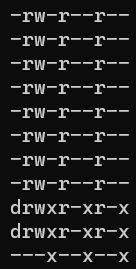
If S- created a file then

u-user/owner - S

g-group - X , Y & Z are the teammates

o-others -5 developers (A B C D E) & 3 Tester



Each File here having 10 bytes out of that

1st - or d : d for directory & - for file

Next 3- for User

Next 3 – for Group

Next 3 – for Others

Always comes in same format

Read- write- execute







1st column: - permission and for file/dir

2nd column: - no. which shows your file/dir is interconnected with how many files/dir

3rd column: - user name

4th column: - Group Name

5th column: -Memory

6th column: - Date/time of creation

7th Column: - dir/ file name

1. chmod
2. Symbolic method

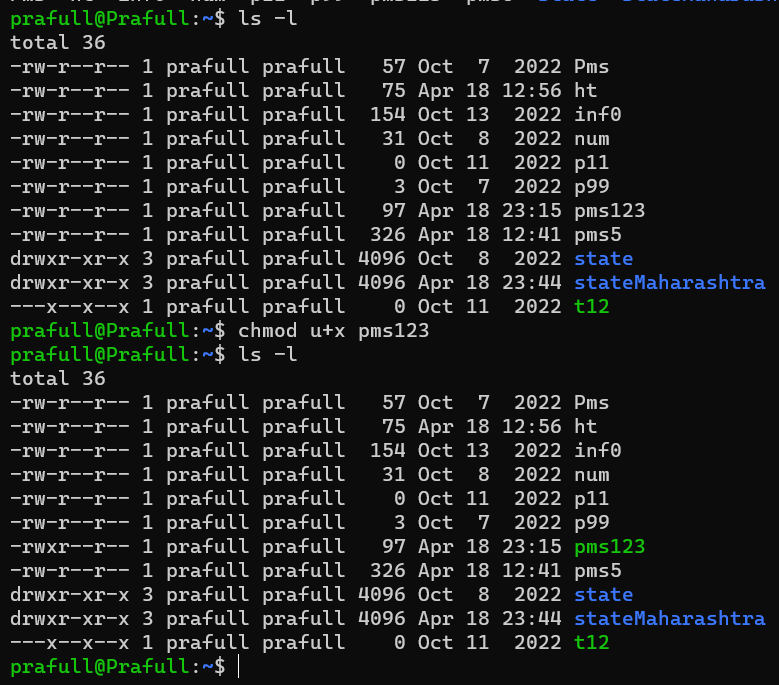
r-read permission

w-write permission

x – execute permission

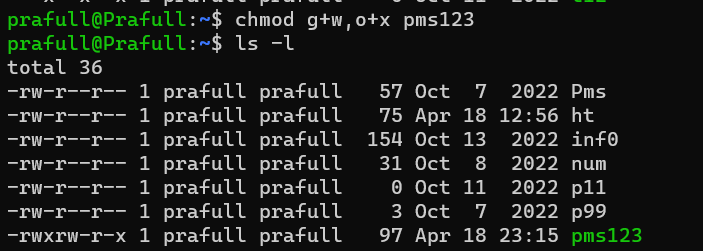
To give Permission use + sign

1. Single Permission

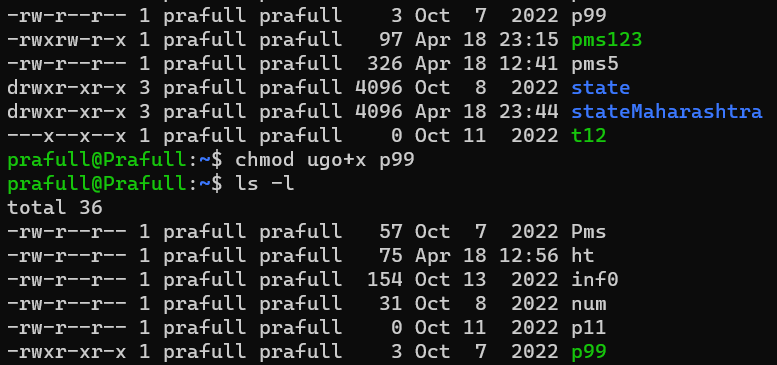




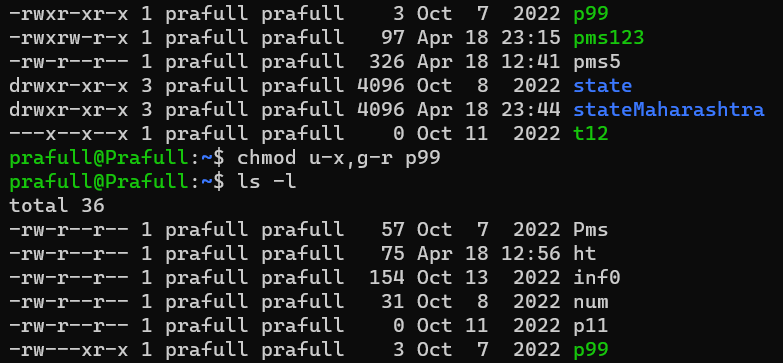
1. Multiple Condition at a time

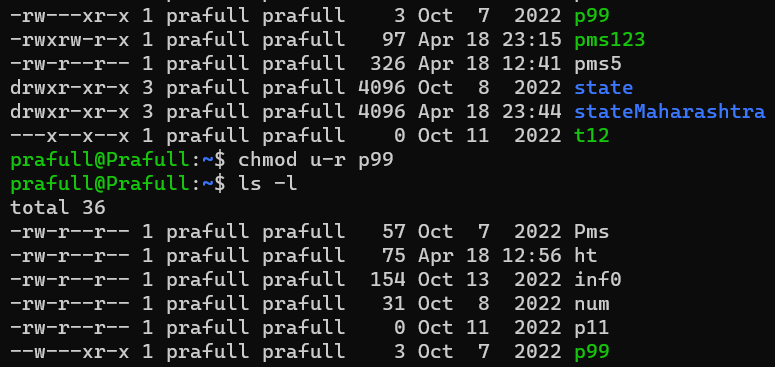


1. For whole group



To remove Permission use – sign





\*\* If we check non-read access file then



1. Number method

r-read permission-4

w-write permission-2

x – execute permission-1

\*\* It will override previous permission

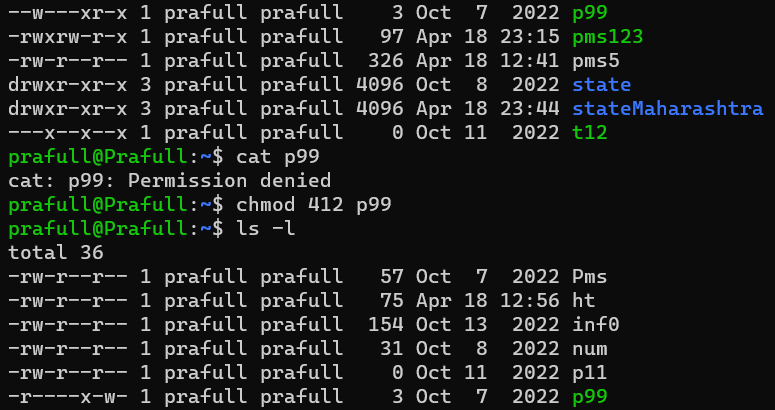
By adding no. we can give multiple condition at a time like below

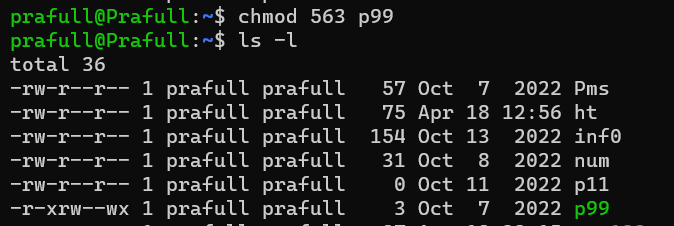
421- user (read access), group (write access), others (execute)

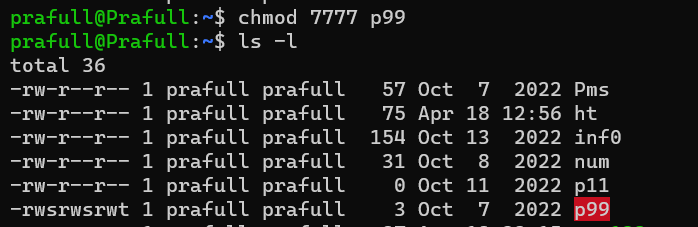
537- user (read+x access), group (write+x access), others (r+w+execute)

777- user (read+w+x access), group (r+write+x access), others (r+w+execute)

645- user (read+w access), group (read access), others (r+execute)







1. Chown (ch own) :-

To give ownership to the other groupmember

Chown smita filename

To get back permission smita needs to run same command by using my name then I’ll get my oweeship back.

1. Chgrp (ch group): - to give permission to whole group

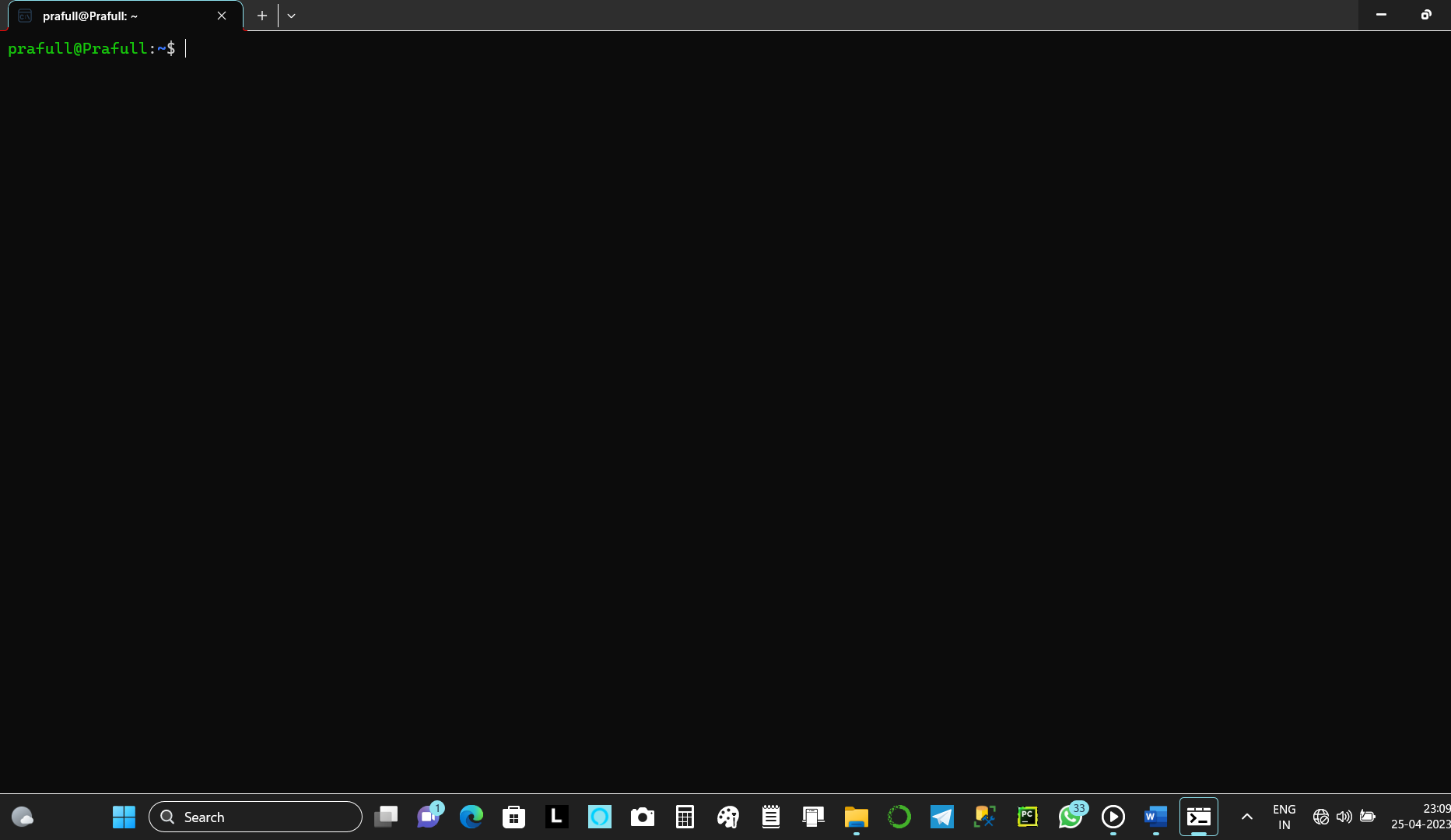
chgrp groupname filename

but even after this command you are going to remain the owner of that group.

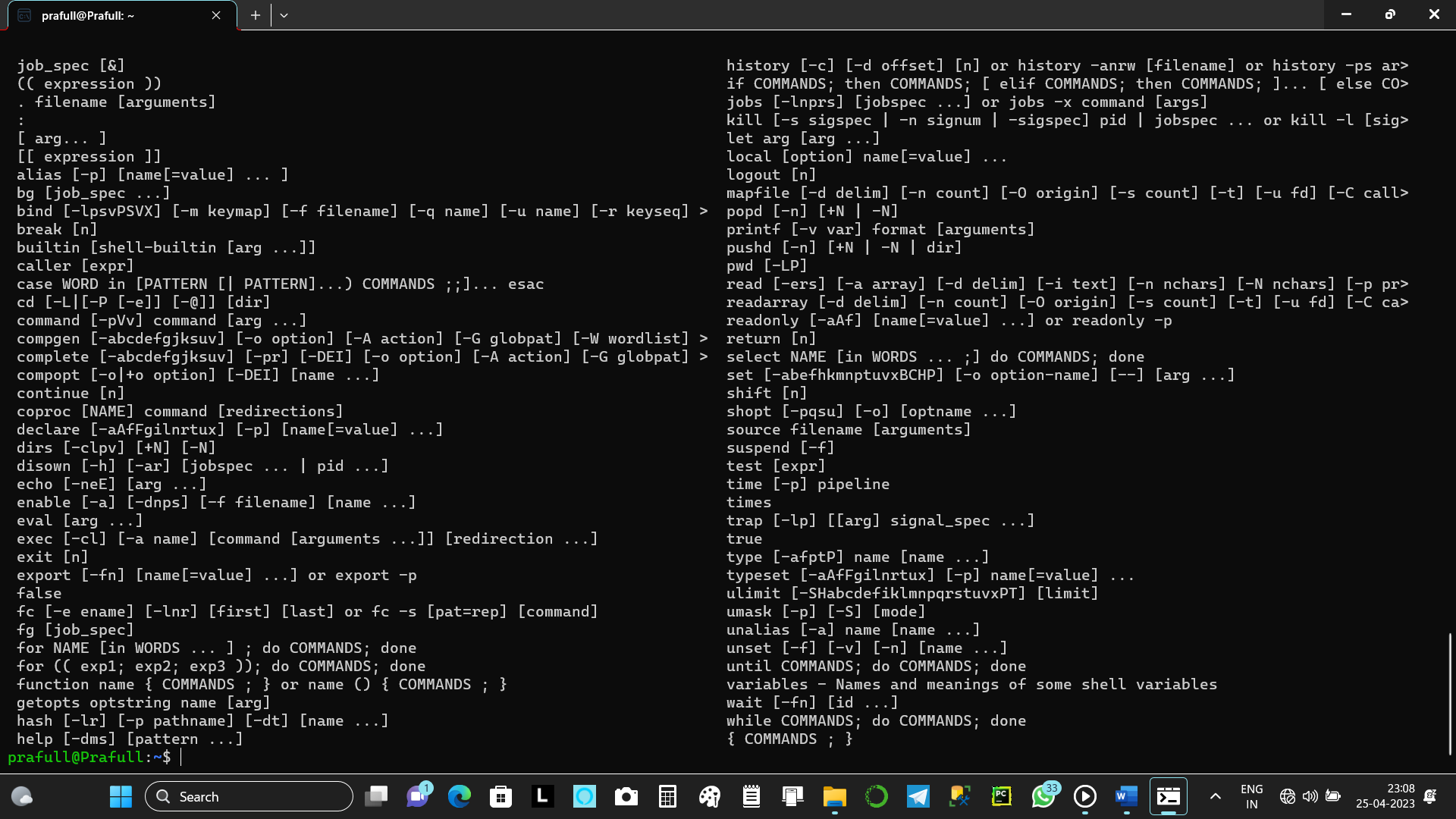
\*\* There is no option for ‘Others’ \*\*

**\*\*Terminal Related Commands \*\***

Clear : - only to clear the page data not to clear history it remains as it is.



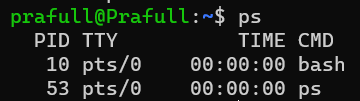
Help: -Show all the data of commands.



**\*\*Process Related Commands \*\***

1. ps – it will show the data of that files that are running at the same time by the group members.

If there is running any process then we have to wait until they drop coz it will coz an error.

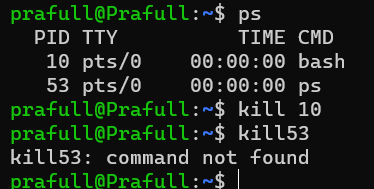


PID- process indicator the no. are random with no meaning

Bash – by default memory file

1. kill PID no. – kill 10

If there is process which takes time to execute at that time we use kill command to terminate that file/dir and re-run that command.



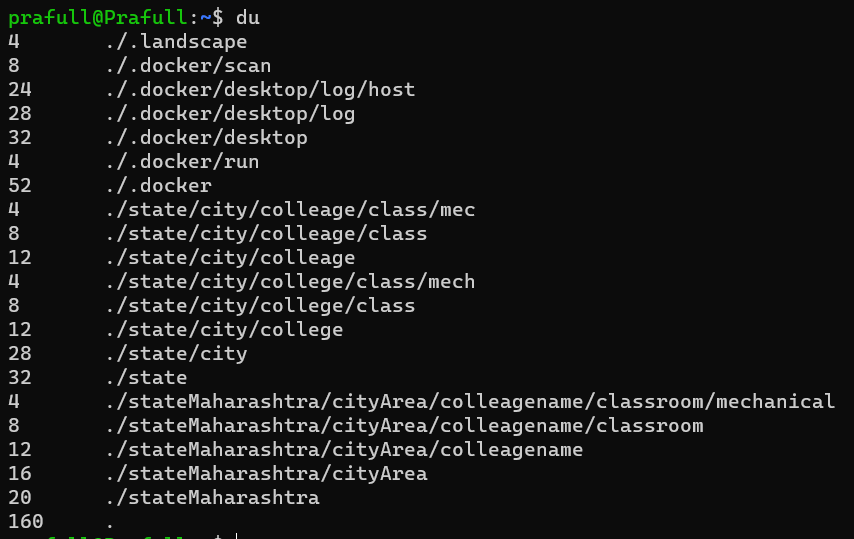
**\*\*Memory Related Commands \*\***

**The result which is shown in memory command we can not say it’s correct coz we are using virtual memory.**

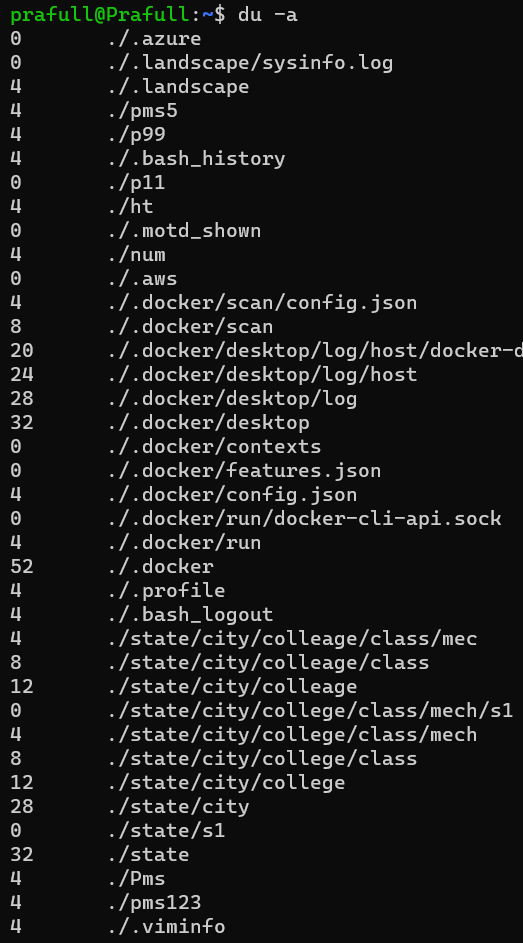
**\* Here we are using virtual memory but in real world we get correct answer\***

1. du (disc usage)[only include dir]

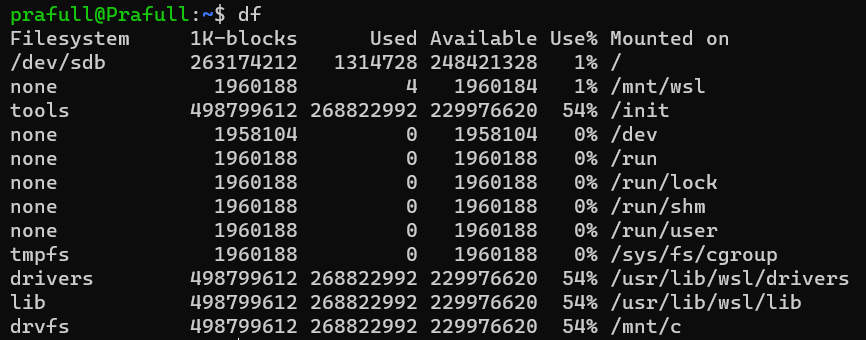
memory used by files



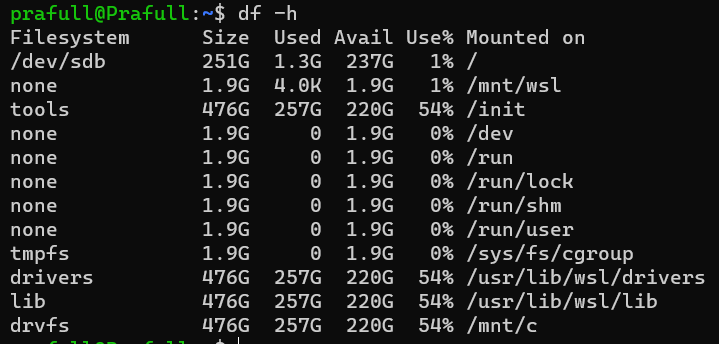
1. du -a : for all dir + file

\

1. df (disc free data available on virtual disc)



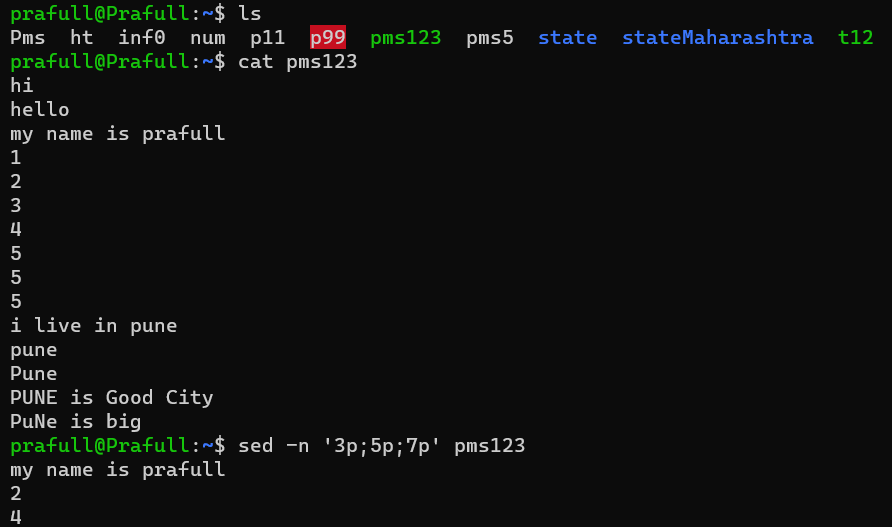
For detailed



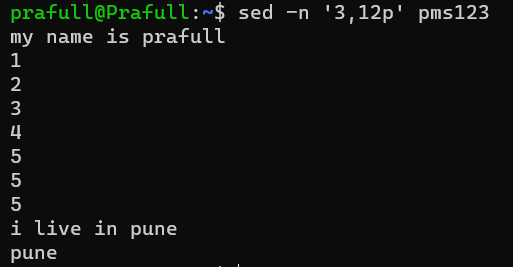
**\*\*Acknowledgement Related Commands \*\***

1. sed (stream editor)
2. To show single lines

sed -n ‘3p;5p;7p’ filename



1. For line no. 3 to 12



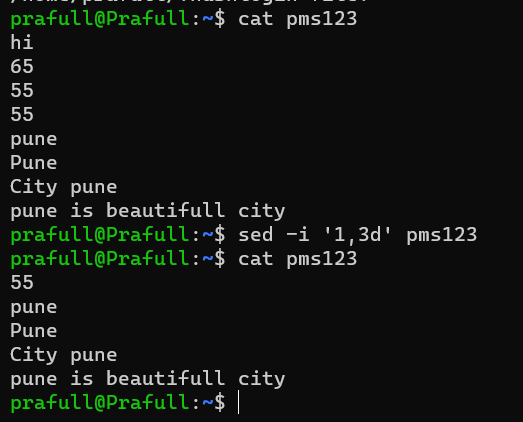
1. Temporary Delete :-



1. Permanent delete :
2. Line by line



1. Multiple line



1. Replace : - case sensitive

sed -i ‘s/1st word/2nd word/g’ filename

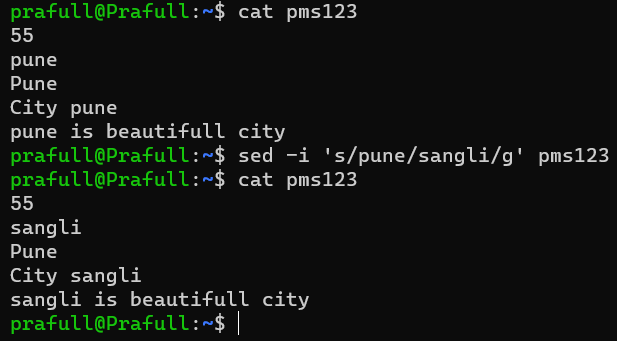
-i -permanent change

S – substitute

1st word – the word which is going to replace

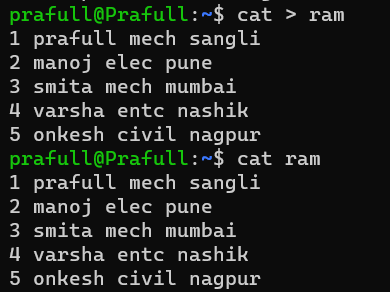
2nd word-the replacement word

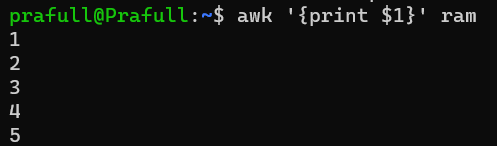
g – global/whole/complete file



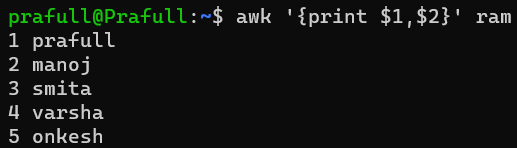
awk – used to fetch column & row

single column



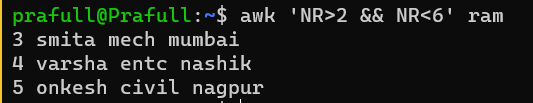


For multiple column



Row (greater than 2 less than 4)



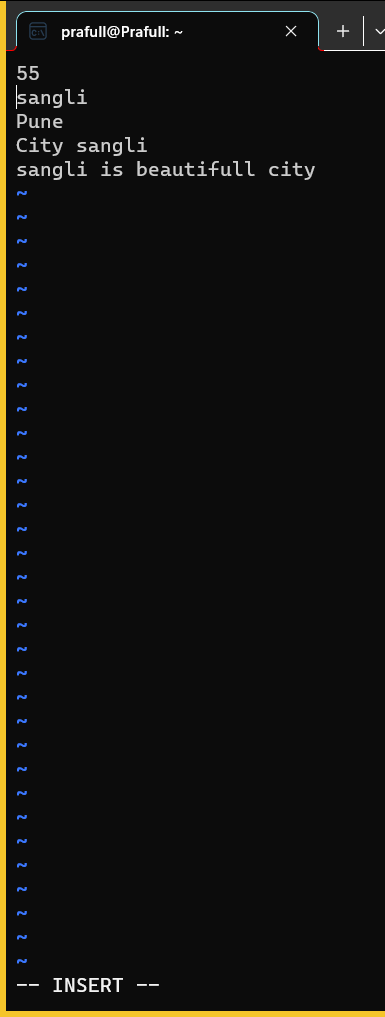


\*\* vi command \*\*

1. To edit file \* vi filename \*



Press i key to insert data – insert mode



Press Esc (escape) to escape/exit from insert mode but without saving the data

K – upside

j – downside

l – right

h - left

x – delete the single character where cursor is

u - undo

X - delete the whole word character where cursor is

dw – delete word where cursor is

dd – delete line where cursor is

number + dd (3dd) – delete 3 line including self but below 2 lines

cut + paste :-

dd- for cut

p-for paste just use cursor to define the location

yy – copy word

yw – copy line

To replace single character –

cursor position + r + press Character to replace

To replace whole word character –

cursor position + R + press word to replace

G – to jump at last position in that whole data

gg - to jump at Starting position in that whole data

dg - delete all data

Esc + :wq! :- save + exit

Esc + w! :- exit without saving the data.